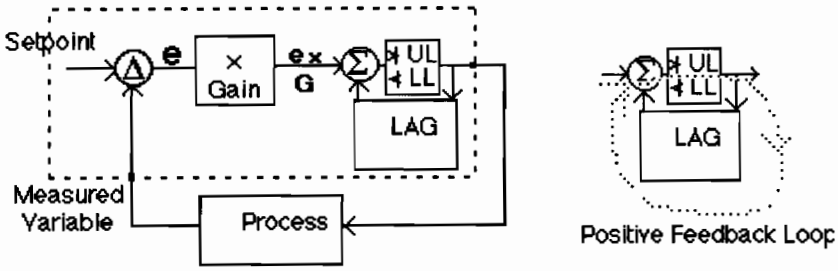
Equivalent code

The PID algorithm, as implemented in a typical digital control system, can be understood by reference to a small basic program that is the equivalent to the PID algorithm in its most common application.

Further explaination of the PID algorithm

1. A positive feedback integral algorithm is used.



2. Derivative is on process.

3. Relative moderate tuning coefficients (<10) are used.

4. Output limits are 0 and 100%

5. The loop is scanned every second

Variables:

`Input` The process input, in percent
`InputD` Process input after derivative calculation
`InputLast` Process input on the previous pass
`InputDF1` Input after derivative calculation and first filter
`InputDF2` Input after derivative calculation and second filter
`Feedback` internal feedback for reset after filter
`Derivative` Derivative time in minutes
`Gain` Gain, negative if controller is reverse acting
`ResetRate` Reset Rate in repeats per minute
`DFilter1` Derivative filter time constants, in minutes.
`DFilter2` These are zero if derivative is not used.
`OutputTemp` Result of the PID calculation
`Output` The final output

The PID emulation code:

```
InputD=Input+(Input-InputLast)*Derivative *60   Derivative calculation
InputLast=Input
InputDF1=InputDF1+(InputD-InputDF1)*DFilter1/60   1st derivative filter
InputDF2=InputDF2+(InputD-InputDF2)*DFilter2/60   2nd derivative filter
OutputTemp=(InputDF2-SetPoint)*Gain+Feedback   Basic gain calculation
IF OutputTemp >100 THEN OutputTemp= 100   Output Limits
IF OutputTemp <0 THEN OutputTemp= 0
Output=OutputTemp   The final output
```

```
Feedback=Feedback+(Feedback-Output)*ResetRate/60
```
Filter for reset feedback

Updated January 31, 1996.
Provided by John Shaw.
Process Control Solutions